WHAT IS CLAIMED IS:

1. A method for building, defining, and storing features in an application neutral format, comprising:

building a feature based on a feature class, wherein the feature class comprises feature geometry, feature constraints, and feature dimensions;

defining the built feature as a geometric representation of an individual feature type; and

storing the representation in a binary file format.

2. The method of claim 1 further comprising ordering the built features.

3. The method of claim 2 further comprising building a three-dimensional model based on the ordering of the built features.

4. The method of claim 3, wherein the three-dimensional model is built in a system utilizing the application neutral format.

5. The method of claim 1 further comprising building a three-dimensional model based on the binary file format.

6. The method of claim 5, wherein the binary file format comprises a geometry library and a feature library adapted to build the three-dimensional model.

7. The method of claim 6, wherein the geometry library comprises geometry classes for:

two-dimensional entities;
three-dimensional entities;
points; and
vectors.

8. The method of claim 7 further comprising copying data between at least one of the class's private data space and an address of the data specified from a calling function.

9. The method of claim 8 further comprising, within each class, classifying the data as at least one of a following classification from a group consisting of:

fundamental data; and
derived data.

10. The method of claim 9 further comprising ensuring, by each of the classes, that any change made to the fundamental data via a function will update the derived data accordingly.

11. The method of claim 1 further comprising independently defining each feature via a three-dimensional coordinate system.

12. The method of claim 11, wherein the three-dimensional coordinate system contains the data necessary to detect at least one of a following element from a group consisting of:

a work plane;
a sketch plane; and
a face upon which a feature may need to be built.

13. The method of claim 12, wherein the data comprises at least one of a following element from a group consisting of:

plane vectors;
an origin of the plane; and
an elevation of the plane from a world origin.

14. The method of claim 1 further comprising inheriting common properties by the feature from a base feature class.

15. The method of claim 14, wherein the base feature class stores data that is common across most feature types.

16. The method of claim 15, wherein each individual feature class contains feature specific data.

17. The method of claim 1 further comprising creating some features by drawing a profile shape from a collection of entities on a specific surface.

18. The method of claim 1 further comprising constraints for the feature.

19. The method of claim 18, wherein the feature constraints are handled via a class that provides at least one of a following action from a group consisting of:

defining a constraint type, a constraint data value, and a constraint object; and

indicating if the constraint is to an edge or to a point, and a definition of the edge or the point, wherein the indicating is based on a constraint object type.

20. The method of claim 1, wherein the binary file format may contain stored two-dimensional input views via a class.

21. The method of claim 20, wherein each view class contains at least one of a following element from a group consisting of:

an array of two-dimensional entities; and

a coordinate system associated with the view.

22. The method of claim 4 further comprising not transferring system specific data through an intermediate file based on the ordering of the built features.

23. The method of claim 4, wherein the system is a Computer Aided Design (CAD) system.

24. The method of claim 1 further comprising not transferring application specific data through an intermediate file based on the ordering of the built features.

25. The method of claim 1, wherein the application is a CAD application.

26. The method of claim 1, wherein the binary file format is a binary file of individual features and metadata associated with each feature.

27. The method of claim 1, wherein the binary file format can be used in a collaborative environment.

28. The method of claim 1, wherein the binary file format can be incrementally updated.

29. A method for incrementally updating a binary file, comprising:

indicating changed features between a first binary file and a second binary file at a first client;

receiving the changed features at a second client;

comparing an identification of the first binary file with the second binary file; and

if the identification is new, writing the second binary file to the first binary file.

30. The method of claim 29 further comprising, prior to the indicating step, reading features by the first client.

31. The method of claim 29 further comprising, prior to the indicating step, writing the features to a binary file format.

32. The method of claim 29 further comprising, prior to the indicating step, comparing features of the second binary file with the first binary file.

33. The method of claim 29 wherein the indicating step comprises marking any changed features.

34. The method of claim 29 further comprising, after the indicating step, writing the changed features to a separate class.

35. The method of claim 29 further comprising, after the indicating step, attaching a unique identifier to the binary file.

36. The method of claim 29 further comprising, after the writing step, updating feature changes in the first binary file.

37. The method of claim 29 further comprising, after the writing step, updating a part related to the feature.

38. The method of claim 37 further comprising redrawing the part if a collaboration between the first client and the second client did not occur.

39. A method for incrementally updating a differential binary file, comprising:

opening a new part by a system that desires to send the part to other collaborating systems;

attaching a tag to the part to identify it in the collaboration;

sending the part for collaboration to a client;

reading the part by the client; and

if the part has a tag attached to its name, searching by the client for another part in the system that has the same tag in its name.

40. The method of claim 39 further comprising, if the other part is not found or there is no tag available for the part, adding a tag to the part.

41. The method of claim 40 further comprising making the part available for collaboration.

42. The method of claim 39 further comprising, if the other part is found, reading the other part and comparing features of the new part with features of the other part.

43. The method of claim 42 further comprising, if a match between features is found, comparing other features until the comparison is complete.

44. The method of claim 42 further comprising, if no match between features is found, adding the new part to a new list until the comparison is complete.

45. The method of claim 44 further comprising, when the comparison is complete, comparing features related to the parts.

46. The method of claim 45 further comprising marking unmatched features and adding them to the new list.

47. The method of claim 46 further comprising comparing the features in the new list.

48. The method of claim 47 further comprising, if the features are the same, deleting the features from the new list.

49. The method of claim 48 further comprising adding a differential binary file tag to the new list.

50. The method of claim 49 further comprising sending a file corresponding to the differential binary file tag to the server for collaboration.

51. The method of claim 50 further comprising sending the file by the server to all other collaborating systems.

52. The method of claim 51 further comprising checking by the other collaborating systems if a differential binary file tag is present.

53. The method of claim 52 further comprising, if the tag is present, mapping the features of the file.

54. The method of claim 53 further comprising receiving a feature type and index from the file.

55. The method of claim 54 further comprising searching the corresponding feature in the collaborating systems.

56. The method of claim 55 further comprising affecting the changes to the part in the collaborating systems.

57. The method of claim 52 further comprising, if the tag is not present, drawing the part in the collaborating systems.

58. A method for converting a two-dimensional drawing to a three-dimensional model, comprising:

detecting the two-dimensional drawing;
correcting errors associated with the two-dimensional drawing;
receiving the corrected drawing by an automated feature detection system;
performing a profile analysis and a feature analysis;
producing a list of three-dimensional features; and
writing the features to a binary file format.

59. The method of claim 58 further comprising interfacing the binary file format to a binary file system.

60. The method of claim 59 further comprising producing a parametric feature-based three-dimensional model.

61. The method of claim 60 further comprising back projecting the three-dimensional model to obtain drawing views associated with a two-dimensional model.

62. The method of claim 61 further comprising overlaying the drawing views on top of the original two-dimensional drawing views.

63. The method of claim 62 further comprising comparing the views.

64. A system for detecting a correcting drawing errors, comprising:

an automated error detection and correction system adapted to receive a two-dimensional drawing; and

a semi-automated error detection and correction system, wherein the automated error detection and correction system is coupled to the semi-automated error detection and correction system;

wherein the automated error detection and correction system is further adapted to automatically detect and correct overlapping entities, duplicate entities, and near zero length entities associated with the drawing; and

wherein the semi-automated error detection and correction system is adapted to receive the resultant drawing, detect disconnected entities and crisscrossing entities associated with the resultant drawing, and produce a corrected two-dimensional drawing.

65. A method for converting a two-dimensional drawing to a three-dimensional model, comprising:

reading a two-dimensional pre-processed drawing;

automatically filtering non-graphical entities;

exploding any blocks in the drawing;

performing error checking on the drawing;

if errors are found, correcting the errors; and

automatically splitting entities in the drawing or in the corrected drawing corresponding to a top, front and side views.

66. The method of claim 65 further comprising fixing a common origin for each view.

67. The method of claim 66 further comprising translating the entities to the common origin.

68. The method of claim 67 further comprising writing the data to classes.

69. A method for detecting a feature associated with a two-dimensional drawing, comprising:

receiving a two-dimensional drawing;
performing a subpart extraction of the drawing;
performing a subpart matching of the drawing;
extracting nested loops and circular loops;
matching the nested loops and circular loops; and
producing matched feature loops.

70. A method for analyzing a feature associated with a drawing, comprising:

receiving matched feature loops;
performing a profile analysis on each loop match;
building feature subtrees;
setting a relative volume operation for each of the feature subtrees;
building feature relations on the feature subtrees;
building a model tree based on the feature relations; and
producing a final feature tree based on the model tree.